UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/538,380	03/29/2000	Jennie Ching	1503P/BC999068	6677

7590 10/06/2003

Sawyer Law Group LLP
P O Box 51418
Palo Alto, CA 94303

EXAMINER	
ALI, SYED J	

ART UNIT	PAPER NUMBER
2127	

DATE MAILED: 10/06/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/538,380	CHING ET AL.	
	Examiner	Art Unit	
	Syed J Ali	2127	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *06 August 2003* .

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-21* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-21* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

1. This office action is in response to Amendment A, paper number 4, which was received August 6, 2003. Applicant's arguments have been fully considered but they deemed to be moot in view of the new ground of rejection. All previous rejections are hereby withdrawn. Claims 1-21 are presented for examination.

2. The text of those sections of Title 35, U.S. code not included in this office action can be found in a prior office action.

Claim Objections

3. Claim 20 is objected to because of the following informalities: The preamble of claim 20 recites, "A computer readable medium containing program instruction..." which appears to be a typographical error. This results in a lack of antecedent basis for further limitations in claim 20 and 21 that refer to "the program instructions". Appropriate correction is required.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claims 1 and 11 are rejected under 35 U.S.C. 102(e) as being anticipated by Sequeira (USPN 6,222,530) (hereinafter Sequeira).

As per claim 1, Sequeira discloses a method for managing subsystem processes from a central site in a digital media distributor (col. 2 lines 48-65, "a distributed broadcast scheduler architecture is disclosed which addresses the deficiencies of prior art program schedulers where devices, such as media servers and tape drives can operate independently of the scheduler", wherein the disclosure is particularly related to the insertion of multimedia in a broadcast via a network connection), the method comprising:

utilizing a plurality of threads as a task manager in a central site server of the digital media distributor (col. 9 lines 16-31, "Threads are used by different parts of the system to perform various computations, functions and tracking. For instance, once a schedule is created, Task Distributor 410 transforms the schedule into a series of tasks and assigns the task to a thread and places the thread in Thread Pool and Queue 430"); and

autonomously controlling initiation and termination of one or more subsystem processes associated with data object transmission of the digital media distributor with the task manager (col. 9 lines 16-31, "At the appropriate time, the thread may issue a command via the Master/Slave Scheduler API 180*a* to Media Server 130 using some network communication mechanism").

As per claim 11, Sequeira discloses a digital media distribution (DMD) system with centralized management of subsystem processes (col. 2 lines 48-65, "a distributed broadcast

scheduler architecture is disclosed which addresses the deficiencies of prior art program schedulers where devices, such as media servers and tape drives can operate independently of the scheduler", wherein the disclosure is particularly related to the insertion of multimedia in a broadcast via a network connection), the DMD system comprising:

a distribution network for data object transmission (col. 6 lines 7-29, "Although shown in Fig. 2 as a single processor system, Computer 210 is not limited to this embodiment and may be, for instance, a multiprocessor system, mainframe, or even a client/server system", wherein a client/server system is of particular relevance to the present claim, and inherently includes a distribution network);

a central site server, the central site server utilizing a plurality of threads as a task manager for autonomous control of initiation and termination of one or more subsystem processes associated with data object transmission of the DMD system (col. 9 lines 16-31, "Threads are used by different parts of the system to perform various computations, functions and tracking. For instance, once a schedule is created, Task Distributor 410 transforms the schedule into a series of tasks and assigns the task to a thread and places the thread in Thread Pool and Queue 430"); and

a plurality of remote site servers for receiving data object transmissions from the central site server via the distribution network (col. 9 line 66 - col. 10 line 12, "In the preferred embodiment, the Master/Slave Scheduler API 130 is a synchronous protocol for distributing to and managing tasks in remote media servers).

Claim Rejections - 35 USC § 103

6. Claims 2 and 12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sequeira in view of Sharma et al. (USPN 5,809,235) (hereinafter Sharma).

As per claim 2, Sharma discloses the following limitations not shown by Sequeira, specifically the method of claim 1 wherein utilizing a plurality of threads further comprises utilizing a main manager thread (col. 21 line 39 – col. 22 line 38, "When the server is first started up, a server management thread is created", wherein the server management thread is responsible for managing the thread pool).

It would have been obvious to one of ordinary skill in the art to combine Sequeira with Sharma since although Sequeira discloses management of subsystem processes through the use of a thread pool, the thread pool appears to have a static number of threads. In certain cases, this could lead to a mismanagement of memory, in particular when considering applications that require high bandwidth, such as media distribution. By utilizing a main manager thread, as disclosed by Sharma, to manage the creation and deletion of threads, threads can be created and destroyed dynamically, thus ensuring that the amount of memory allocated to the thread pool suits the needs of the system at any particular point in time.

As per claim 12, Sharma discloses the following limitations not shown by Sequeira, specifically the system of claim 11 wherein the central site server utilizes a main manager thread for the task manager (col. 21 line 39 – col. 22 line 38, "When the server is first started up, a

server management thread is created", wherein the server management thread is responsible for managing the thread pool).

The motivation for combining Sequeira and Sharma is provided above in reference to claim 2.

7. Claims 3 and 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sequeira in view of Sharma in view of Dangelo et al. (USPN 5,907,494) (hereinafter Dangelo).

As per claim 3, Dangelo discloses the following limitations not shown by the modified Sequeira, specifically the method of claim 2 wherein utilizing a plurality of threads further comprises utilizing a subsystem control thread as a child thread of the main manager thread (col. 6 lines 24-34, "the virtual machine 55 enables peer point tools to coordinate their activities or to spawn new 'child' activities via interprocess control threads", wherein Dangelo discloses the ability to create child threads in an effort to control the execution of processes on the system).

It would have been obvious to one of ordinary skill in the art to combine the modified Sequeira with Dangelo since by allowing the execution of processes to be maintained by control threads rather than by the main manager thread, each type of thread can have a more specialized function. That is, the main manager thread is responsible for only creating and deleting threads as necessary, while the child threads manage memory allocation, execution of system tasks, etc. Additionally, the idea presented by Dangelo of spawning "child" threads to manage subtasks of the system is well established in the art, and is explained in detail by Dangelo.

As per claim 13, Dangelo discloses the following limitations not shown by the modified Sequeira, specifically the system of claim 12 wherein the central site server utilizes a subsystem control thread as a child thread of the main manager thread (col. 6 lines 24-34, "the virtual machine 55 enables peer point tools to coordinate their activities or to spawn new 'child' activities via interprocess control threads", wherein Dangelo discloses the ability to create child threads in an effort to control the execution of processes on the system).

The motivation for combining the modified Sequeira with Dangelo is provided above in reference to claim 3.

8. Claims 4-6, 8-9, 14-16, and 18-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sequeira in view of Sharma in view of Dangelo in view of Guedalia et al. (USPN 6,535,878) (hereinafter Guedalia).

As per claim 4, Guedalia discloses the following limitations not shown by the modified Sequeira, specifically the method of claim 3 wherein utilizing a plurality of threads further comprises utilizing a watchdog worker thread as a child thread of the subsystem control thread (col. 10 lines 12-44, "a special 'watchdog' thread is used to monitor the threads", wherein the watchdog manages creation and deletion of threads that ultimately perform all processes of the system).

It would have been obvious to one of ordinary skill in the art to combine the modified Sequeira with Guedalia since it provides a way of monitoring thread creation and deletion, thereby providing programming flexibility as well as simpler memory management. The

watchdog thread is used to create and delete threads at regular intervals, thereby keeping the number of threads in the thread pool constant, while also ensuring that all threads are given an adequate amount of service. This prevents common problems associated with threads, such as deadlock and starvation.

As per claim 5, Guedalia discloses the method of claim 4 wherein utilizing a plurality of threads further comprises utilizing a spawn worker thread as a child thread of the watchdog worker thread (col. 10 lines 12-44, "the 'watchdog' manages thread creation. Whenever the 'watchdog' finds, during its regular check, that the tick counter of a thread has reached 3, it then lowers the priority of this thread and removes it from the thread pool, and creates a new thread to replace it", wherein the watchdog thread spawns a new thread in response to a work request and the spawned worker thread is added to the thread pool).

As per claim 6, Sharma discloses the method of claim 5 further comprising utilizing one watchdog worker thread and one spawn worker thread for each subsystem process (col. 21 line 39 – col. 22 line 38, "there is a one to one correspondence between a TAccess Definition Object with each client_port assigned on the server side", wherein a server thread is allocated from the thread pool for each work request).

It is noted that Sharma does not include watchdog threads for the management of each subsystem process. However, as discussed regarding claims 4 and 5, Guedalia teaches that a watchdog thread manages all of the thread creation of the system. This suggests that each

subsystem process would then have one watchdog worker thread overseeing its execution, as claimed.

As per claim 8, Guedalia discloses the method of claim 6 further comprising utilizing the watchdog worker thread to start each subsystem process through the spawn worker thread and to monitor performance of each subsystem process (col. 10 lines 12-44, "a special 'watchdog' thread is used to monitor the threads", "the 'watchdog' manages thread creation", wherein the watchdog thread is solely responsible for creation and deletion of threads and also monitors their execution in order to update their status).

As per claim 9, Guedalia discloses the method of claim 8 wherein utilizing the spawn worker thread further comprises spawning each subsystem process and waiting for termination of each spawned subsystem process (col. 10 lines 12-44, "the 'watchdog' thread manages thread creation", "The old thread that was removed from the thread pool completes its task and dies", wherein the watchdog thread creates a thread of execution and removes it from the thread pool upon its completion of its task).

As per claim 14, Guedalia discloses the following limitations not shown by the modified Sequeira, specifically the system of claim 13 wherein the central site server utilizes a watchdog worker thread as a child thread of the subsystem control thread (col. 10 lines 12-44, "a special 'watchdog' thread is used to monitor the threads", wherein the watchdog manages creation and deletion of threads that ultimately perform all processes of the system).

The motivation for combining the modified Sequeira with Guedalia is provided above in reference to claim 4.

As per claim 15, Guedalia discloses the system of claim 14 wherein the central site server utilizes a spawn worker thread as a child thread of the watchdog worker thread (col. 10 lines 12-44, "the 'watchdog' manages thread creation. Whenever the 'watchdog' finds, during its regular check, that the tick counter of a thread has reached 3, it then lowers the priority of this thread and removes it from the thread pool, and creates a new thread to replace it", wherein the watchdog thread spawns a new thread in response to a work request and the spawned worker thread is added to the thread pool).

As per claim 16, Sharma discloses the system of claim 15 wherein the central site server utilizes one watchdog worker thread and one spawn worker thread for each subsystem process (col. 21 line 39 – col. 22 line 38, "there is a one to one correspondence between a TAccess Definition Object with each client_port assigned on the server side", wherein a server thread is allocated from the thread pool for each work request).

It is noted that Sharma does not include watchdog threads for the management of each subsystem process. However, as discussed regarding claims 4 and 5, Guedalia teaches that a watchdog thread manages all of the thread creation of the system. This suggests that each subsystem process would then have one watchdog worker thread overseeing its execution, as claimed.

As per claim 18, Guedalia discloses the system of claim 16 wherein the central site server further utilizes the watchdog worker thread to start each subsystem process through the spawn worker thread and to monitor performance of each subsystem process (col. 10 lines 12-44, "a special 'watchdog' thread is used to monitor the threads", "the 'watchdog' manages thread creation", wherein the watchdog thread is solely responsible for creation and deletion of threads and also monitors their execution in order to update their status).

As per claim 19, Guedalia discloses the system of claim 18 wherein the central site server further utilizes the spawn worker thread for spawning each subsystem process and waiting for termination of each spawned subsystem process (col. 10 lines 12-44, "the 'watchdog' thread manages thread creation", "The old thread that was removed from the thread pool completes its task and dies", wherein the watchdog thread creates a thread of execution and removes it from the thread pool upon its completion of its task).

9. Claims 7 and 17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sequeira in view of Sharma in view of Dangelo in view of Flenley et al. (USPN 6,405,317) (hereinafter Flenley).

As per claim 7, Flenley discloses the following limitations not shown by the modified Sequeira, specifically the method of claim 3 further comprising utilizing the subsystem control thread to determine need for initiation of a subsystem process (col. 4 lines 11 –32, "Each instantiation spawns its own control thread that is responsible for monitoring its own private

queue....if there is a request pending execution and it is time to attempt to process pending requests again, the SPM objects spawns an execution thread that handles the execution of that single request", wherein the control thread disclosed by Flenley spawns an execution thread in response to a need to service a particular request).

It would have been obvious to one of ordinary skill in the art to combine the modified Sequeira with Flenley since by making the control thread determine when an execution thread needs to be created, the system further breaks down tasks into smaller, more manageable tasks. The control thread thereby allocates an execution thread (similar to the thread pool discussed above) to handle any service requests, thereby creating a set of thread types, each with a very specific function, to modularize the functionality of the system, and thereby make it more manageable and customizable.

As per claim 17, Flenley discloses the following limitations not shown by the modified Sequeira, specifically the system of claim 13 wherein the central site server further utilizes the subsystem control thread to determine need for initiation of a subsystem process (col. 4 lines 11 –32, "Each instantiation spawns its own control thread that is responsible for monitoring its own private queue....if there is a request pending execution and it is time to attempt to process pending requests again, the SPM objects spawns an execution thread that handles the execution of that single request", wherein the control thread disclosed by Flenley spawns an execution thread in response to a need to service a particular request).

The motivation for combining the the modified Sequeira with Flenley is provided above in reference to claim 7.

10. Claim 10 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sequeira in view of Wang (USPN 5,758,645).

As per claim 10, Wang discloses the following limitations not shown by Sequeira, specifically the method of claim 1 wherein controlling one or more subsystem processes further comprises controlling a subsystem process from the group comprising a scheduler process, a stage manager process, a local insertion system proxy process, an error document check process, a response document processor process, a disk pool manager process, a request generator process, As-Run manager processes, an update network break time process, and a network local broadcast process (col. 3 line 57 – col. 4 line 10, "This layer is over the IBM Parallel System Support Programs [PSSP] layer", "The PSSP layer 28 is over the AIX operating system layer", wherein Applicant discloses on pg. 8 that the claimed subsystem processes are part of the operating system of Fig. 2 element 34, and the Figure clarifies the operating system as being a combination of AIX and PSSP).

It would have been obvious to one of ordinary skill in the art to combine Sequeira with Wang since the processes being executed must be a set of processes an operating system requests service for. The PSSP and AIX definitions incorporated into Wang show that these are two operating systems that commonly are used together, thus a need exists for a system that services the processes of these systems. Sequeira provides an exemplary model of a scheduler that services such tasks, by providing a task manager that assigns a thread from a thread pool to a

particular task, and schedules events that are specifically related to media servers, as they relate to broadcast scheduling.

11. Claim 20 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sequeira.

As per claim 20, Sequeira discloses a computer readable medium containing program instruction for managing subsystem processes from a central site in a digital media distributor (DMD) system (col. 2 lines 48-65, "a distributed broadcast scheduler architecture is disclosed which addresses the deficiencies of prior art program schedulers where devices, such as media servers and tape drives can operate independently of the scheduler", wherein the disclosure is particularly related to the insertion of multimedia in a broadcast via a network connection), the program instructions comprising:

providing a task manager as a main program thread of an operating system of a central site server of the DMD system (col. 9 lines 16-31, "Threads are used by different parts of the system to perform various computations, functions and tracking. For instance, once a schedule is created, Task Distributor 410 transforms the schedule into a series of tasks and assigns the task to a thread and places the thread in Thread Pool and Queue 430"; see also Fig. 3B, elements 170b and 180a).

"Official Notice" is taken that managing subsystem processes associated with data object transmissions of the DMD system from start-up to shut down, includes states of online, offline, process inoperable, deadlock inoperable, and spawn inoperable, with the task manager to dynamically manage the DMD system is well known and expected in the art. Specifically, the

above states of a server are well known possible statuses of a server at any given time. Further, there are many methods of troubleshooting such a server. Sequeira discloses a way of automating the activities of a server based on a main managing thread, and it would easily fall within the scope of Sequeira to state that certain states of operation of a thread are watched over by the task manager. Although Sequeira does not specify that what is disclosed must be utilized in a client/server model, Sequeira does specify that is one of the possible frameworks for the method disclosed therein. If a networking environment were to be used, server troubleshooting would have to be implemented. In view of this, many troubleshooting techniques for servers exist, several of which would be able to handle the claimed system states.

12. Claim 21 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sequeira in view of Dangelo in view of Guedalia.

As per claim 21, Dangelo discloses the following limitations not shown by Sequeira, specifically the program instructions of claim 20 wherein providing a task manager further comprises utilizing a control thread (col. 6 lines 24-34, "the virtual machine 55 enables peer point tools to coordinate their activities or to spawn new 'child' activities via interprocess control threads", wherein Dangelo discloses the ability to create child threads in an effort to control the execution of processes on the system).

It would have been obvious to one of ordinary skill in the art to combine Sequeira with Dangelo since specializing thread creation through use of a control thread allows greater programming flexibility. Specifically, the control threads can handle threads of execution within

a single process, and also monitor all interprocess communication for those threads. This allows memory management to be more localized, thereby providing fewer leaks and better protection against data faults.

Guedalia discloses the following limitations not shown by the modified Sequeira, specifically the program instructions of claim 20 providing a task manager further comprises utilizing worker threads for managing the subsystem processes (col. 10 lines 12-44, "a special 'watchdog' thread is used to monitor the threads", wherein the watchdog manages creation and deletion of threads that ultimately perform all processes of the system).

It would have been obvious to one of ordinary skill in the art to combine the modified Sequeira with Guedalia since having an individual worker thread for each process allows specialization in thread types, such that each type of worker thread can be associated with a specific type of process. Particularly in relation to a DMD system, wherein each thread may be associated with a different media type, specialized local methods may be defined for each thread type by the programmer, thereby providing increased functionality.

Conclusion

13. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after

the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Syed J Ali whose telephone number is (703) 305-8106. The examiner can normally be reached on Mon-Fri 8-5:30, 2nd Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William A Grant can be reached on (703) 308-1108. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Syed Ali
September 26, 2003

WILLIAM GRANT
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

9/30/03